Reuters Corpus

Jason Antal

2024-08-19

Revisit the Reuters C50 text corpus that we briefly explored in class. Your task is simple: tell an interesting story, anchored in some analytical tools we have learned in this class, using this data. For example:

```
you could cluster authors or documents and tell a story about what you find.
you could look for common factors using PCA.
you could train a predictive model and assess its accuracy, constructing features for each document tha
you could do anything else that strikes you as interesting with this data.
```

Describe clearly what question you are trying to answer, what models you are using, how you pre-processed the data, and so forth. Make sure you include at least one really interesting plot (although more than one might be necessary, depending on your question and approach.)

Format your write-up in the following sections, some of which might be quite short:

```
Question: What question(s) are you trying to answer?
Approach: What approach/statistical tool did you use to answer the questions?
Results: What evidence/results did your approach provide to answer the questions? (E.g. any numbers, ta
Conclusion: What are your conclusions about your questions? Provide a written interpretation of your re
```

Regarding the data itself: In the C50train directory, you have 50 articles from each of 50 different authors (one author per directory). Then in the C50test directory, you have another 50 articles from each of those same 50 authors (again, one author per directory). This train/test split is obviously intended for building predictive models, but to repeat, you need not do that on this problem. You can tell any story you want using any methods you want. Just make it compelling!

Note: if you try to build a predictive model, you will need to figure out a way to deal with words in the test set that you never saw in the training set. This is a nontrivial aspect of the modeling exercise. (E.g. you might simply ignore those new words.)

This question will be graded according to three criteria:

```
the overall "interesting-ness" of your question and analysis.
the clarity of your description. We will be asking ourselves: could your analysis be reproduced by a co
technical correctness (i.e. did you make any mistakes in execution or interpretation?)
```

```r
cat('Our question for this analysis is: Are Reuters stories more positive in tone,
negative, or neutral?')
```

```
## Our question for this analysis is: Are Reuters stories more positive in tone,
## negative, or neutral?
```

```r
cat('To start, we will extract all content in our folder of 2500 stories
      using the PlainTextDocument() function.')
```

```
## To start, we will extract all content in our folder of 2500 stories
##       using the PlainTextDocument() function.
```

```r
extract_content <- function(fname) {
  tryCatch({
    content <- readLines(fname, warn = FALSE, encoding = "UTF-8")
    content <- paste(content, collapse = "\n")

    return(PlainTextDocument(
      x = content,
      id = fname,
      language = 'en'
    ))})}

# Read all files and extract content
processed_files <- lapply(seq_along(train_file_list), function(i) {
  if (i %% 100 == 0)
  extract_content(train_file_list[i])
})

cat('We will now preprocess the data by getting rid of uninformative words like
"the", "and", and "character". We will also remove any tags in the files such as
"datetimestamp". Once we have done this, we will use several built-in functions to
clean the text, such as lowercasing and stripping white space.')
```

```
## We will now preprocess the data by getting rid of uninformative words like
## "the", "and", and "character". We will also remove any tags in the files such as
## "datetimestamp". Once we have done this, we will use several built-in functions to
## clean the text, such as lowercasing and stripping white space.
```

```r
extended_stopwords <- c(stopwords("english"), "and", "the", "for", "that", "with", "was", "said", "from
                        "will", "also", "to", "it", "there", "is", "my", "can", "I", "this", "a", "of",
                        "character", "billion", "gmt", "hour", "mday", "wday", "yday", "one", "last", "
                        "first", "group", "years", "analyst", "listsec", "told", "may", "mon", "percent
                        "government", "told", "heading", "zone", "origin", "description", 'null', 'year
                        "datetimestamp", "gmtoff", "isdst", "listcontent", "meta", "listauthor")

corpus <- Corpus(VectorSource(processed_files))

# Preprocess the text
corpus <- corpus %>%
  tm_map(content_transformer(tolower)) %>%
  tm_map(removePunctuation) %>%
  tm_map(removeNumbers) %>%
  tm_map(removeWords, extended_stopwords) %>%
  tm_map(stripWhitespace)
```

```
## Warning in tm_map.SimpleCorpus(., content_transformer(tolower)): transformation
## drops documents
## Warning in tm_map.SimpleCorpus(., removePunctuation): transformation drops
## documents

## Warning in tm_map.SimpleCorpus(., removeNumbers): transformation drops
## documents

## Warning in tm_map.SimpleCorpus(., removeWords, extended_stopwords):
## transformation drops documents

## Warning in tm_map.SimpleCorpus(., stripWhitespace): transformation drops
## documents
```

```r
# Create a document-term matrix
dtm <- DocumentTermMatrix(corpus)

cat('We now calculate the most frequent terms and output them in a wordcloud.')
```

```
## We now calculate the most frequent terms and output them in a wordcloud.
```

```r
# Calculate term frequencies
term_freq <- colSums(as.matrix(dtm))

# Get the most common terms
top_25_terms <- sort(term_freq, decreasing = TRUE)[1:25]
cat("Top 25 most common terms:")
```

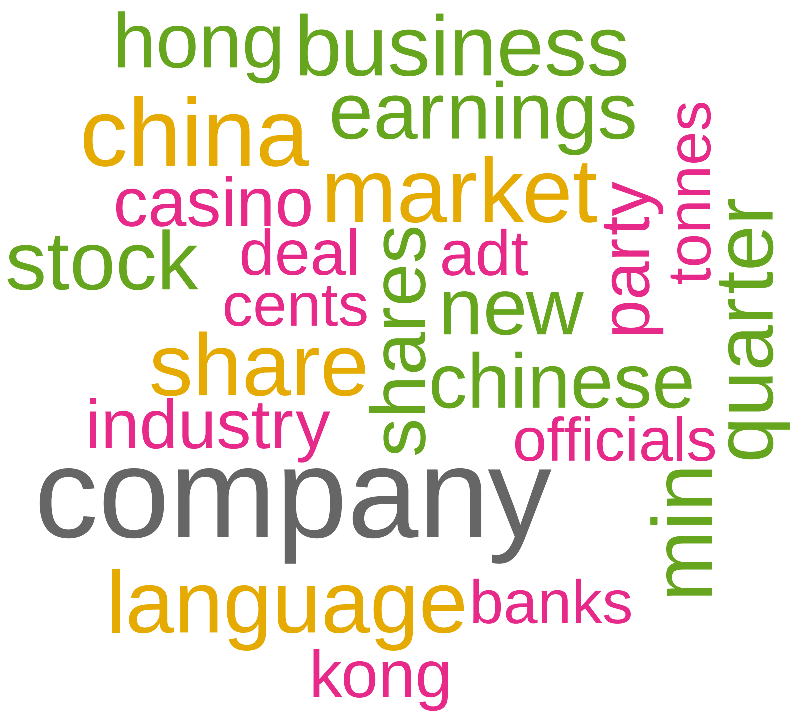
```
## Top 25 most common terms:
```

```r
print(top_25_terms)
```

```
##   company  analysts     china    market  language     share       min  business
##        40        35        29        27        26        26        25        25
##     stock   quarter       new  earnings    shares   chinese      hong     party
##        24        24        23        23        22        22        22        19
##    casino  industry      kong      deal       adt officials     banks     cents
##        19        19        18        17        17        16        16        16
##    months
##        16
```

```r
wordcloud(names(term_freq), term_freq, max.words = 25, colors = brewer.pal(8, "Dark2"))
```

```
## Warning in wordcloud(names(term_freq), term_freq, max.words = 25, colors =
## brewer.pal(8, : analysts could not be fit on page. It will not be plotted.
```



```
cat('We can see that the most common terms are generally focused on financial
      markets and trade, and not particularly positive or negative. This was interesting,
      but it does not answer our original question. So to understand
      which way Reuters leans in terms of tone, we will perform a sentiment analysis.')
```

```
## We can see that the most common terms are generally focused on financial
##       markets and trade, and not particularly positive or negative. This was interesting,
##       but it does not answer our original question. So to understand
##       which way Reuters leans in terms of tone, we will perform a sentiment analysis.
```

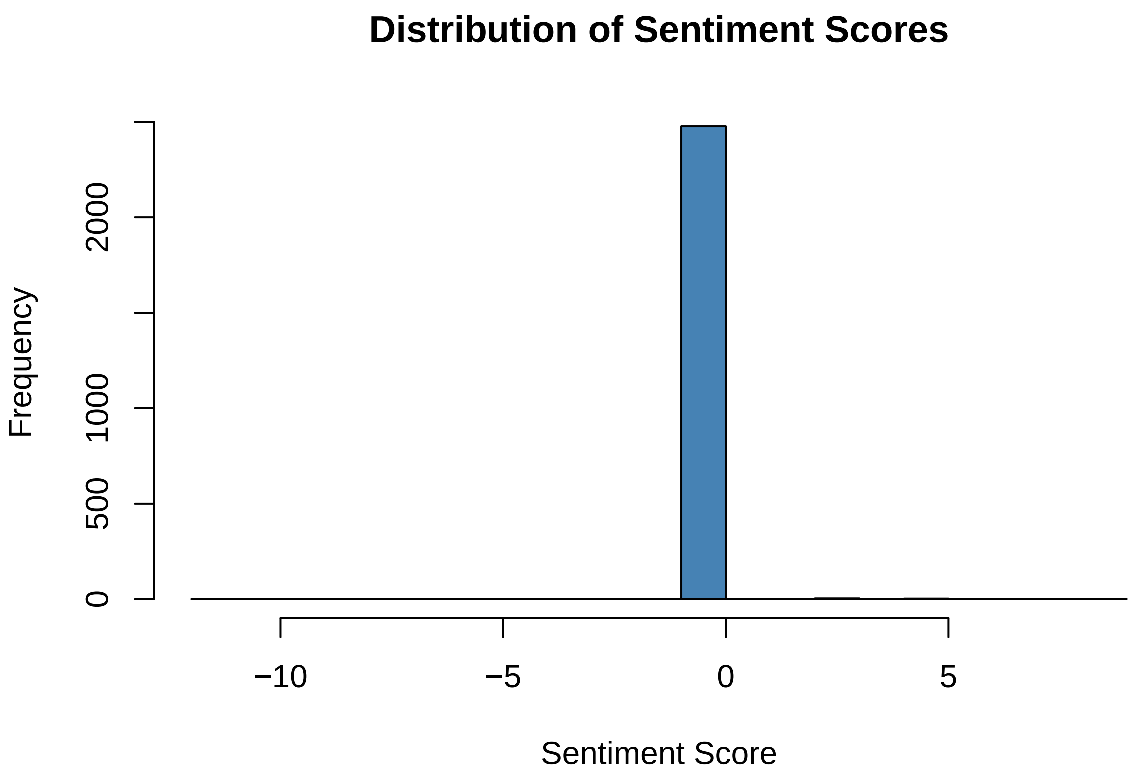
```
analyze_sentiment <- function(doc) {
  text <- as.character(doc)
  sentiment <- get_sentiment(text, method="bing")
  return(sum(sentiment))
}

# Perform sentiment analysis on all documents
sentiment_scores <- sapply(corpus, analyze_sentiment)
overall_sentiment <- mean(sentiment_scores)

# Print results
cat("Overall sentiment score:", overall_sentiment, "\n")
```

```
## Overall sentiment score: 0.01
```

```
print(table(sentiment_scores))
```

```
## sentiment_scores
##  -12   -7   -6   -5   -4   -3   -1    0    1    2    3    4    5    7    9
##    1    1    1    1    2    1    1 2477    2    1    4    1    3    2    2
```

```r
hist(sentiment_scores, main="Distribution of Sentiment Scores",
     xlab="Sentiment Score", ylab="Frequency", breaks=20, col = "steelblue")
```



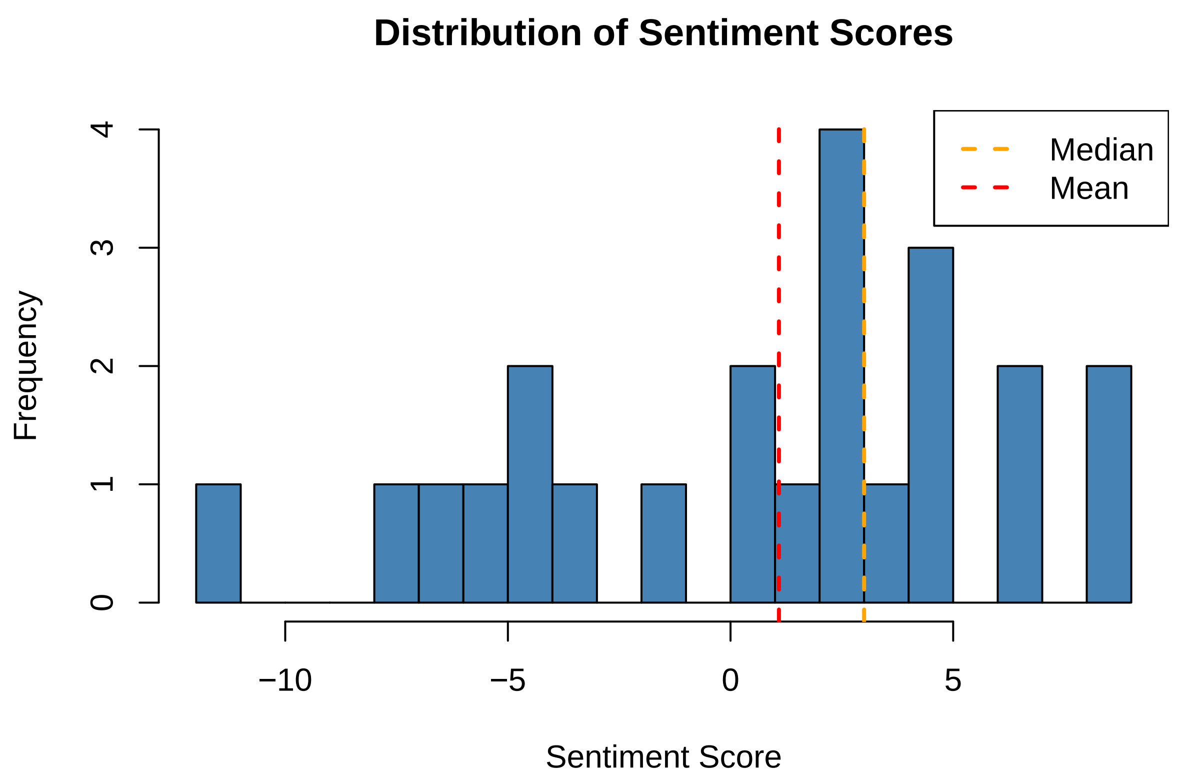
```r
cat("This barchart is not an error; when we break down the results, we can
      see that 2,477 articles were ranked as more or less neutral in tone. It seems
      the people at Reuters are doing a good job at being impartial in their writing.")
```

```
## This barchart is not an error; when we break down the results, we can
##       see that 2,477 articles were ranked as more or less neutral in tone. It seems
##       the people at Reuters are doing a good job at being impartial in their writing.
```

```r
cat("For the sake of answering our original question of whether or not
      Reuters stories trend negative or positive, let's repeat the process
      with all neutral scores removed.")
```

```
## For the sake of answering our original question of whether or not
##       Reuters stories trend negative or positive, let’s repeat the process
##       with all neutral scores removed.
```

```r
sentiment_scores2 <- sentiment_scores[sentiment_scores != 0]

hist(sentiment_scores2, main="Distribution of Sentiment Scores",
     xlab="Sentiment Score", ylab="Frequency", breaks=20, col = "steelblue")
abline(v=median(sentiment_scores2), col="orange", lwd=2, lty=2)
abline(v=mean(sentiment_scores2), col="red", lwd=2, lty=2)
legend("topright", legend=c("Median", "Mean"), col=c("orange", "red"), lty=2, lwd=2)
```



```
cat("From this graph, we can see that the few Reuters stories that are distinctly
      positive or negative tend to trend positive.")
```

```
## From this graph, we can see that the few Reuters stories that are distinctly
##       positive or negative tend to trend positive.
```